JavaScript Promises

Thinking Sync in an Async World

then

Kerrick Long

Things I make and do

meetup.com/**STLEmber**

Lead Front-end Developer
at Second Street

Where to find me online

www.**KerrickLong**.com

twitter.com/**KerrickLong**

github.com/**Kerrick**

```
try {
    console.log(getLatestGrade(getStudent(name)));
}
catch (error) {
    logError(error);
}
finally {
    logOut();
}
```

```
try {
    console.log(getLatestGrade(getStudent(name)));
}
catch (error) {
    logError(error);
}
finally {
    logOut();
}
```

XMLHttpRequest

```
getStudent(name, function(student) {
    getLatestGrade(student, function(grade) {
        console.log(grade);
        logOut();
    });
});
```

```
var handleError = function(error) {
    logError(error);
    logOut();
};
getStudent(name, function(error, student) {
    if (error) return handleError(error);
    getLatestGrade(student, function(error, grade) {
        if (error) return handleError(error);
        console.log(grade);
        logOut();
    });
});
```

```
var handleError = function(error) {
    logError(error);
    logOut();
};
getStudent(name, {
    error: handleError,
    success: function(student) {
        getLatestGrade(student, {
            error: handleError
            success: function(grade) {
                console.log(grade);
                logOut();
            },
        })
    },
});
```

```
try {
    console.log(getLatestGrade(getStudent(name)));
}
catch (error) {
    logError(error);
}
finally {
    logOut();
}
```

Awesome

```
var handleError = function(error) {
    logError(error);
    logOut();
};
getStudent(name, {
    error: handleError,
    success: function(student) {
        getLatestGrade(student, {
            error: handleError
            success: function(grade) {
                console.log(grade);
                logOut();
            },
        })
    },
});
```

Awkward.

```
getStudent(name)
    .then(getLatestGrade)
    .then(console.log)
    .catch(logError)
    .then(logOut)
;
```

Promises

Promise Basics

Promise Basics

Consuming Promises

Promise Basics

Consuming Promises

Creating Promises

Promise Basics

Consuming Promises

Creating Promises

Advanced Techniques

Promise Basics

Getting Promises

RSVP.js

Q.js • Bluebird

RSVP.js

Q.js • Bluebird

Promises/A+

Promise Guarantees

Always Async

Always Async

Returns a Promise

Always Async

Returns a Promise

Handled Once

Always Async

Returns a Promise

Handled Once

Then’able

Promise States

Promise States

Pending **Fulfilled** **Rejected**

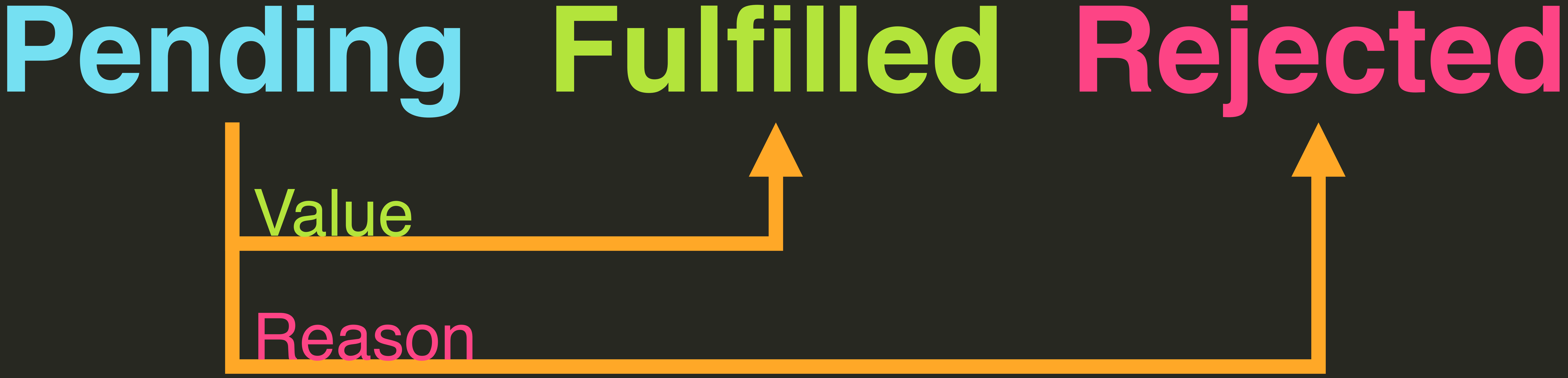
Promise States

$$\text{Pending} \quad \underbrace{\text{Fulfilled} \quad \text{Rejected}}_{\text{Settled}}$$

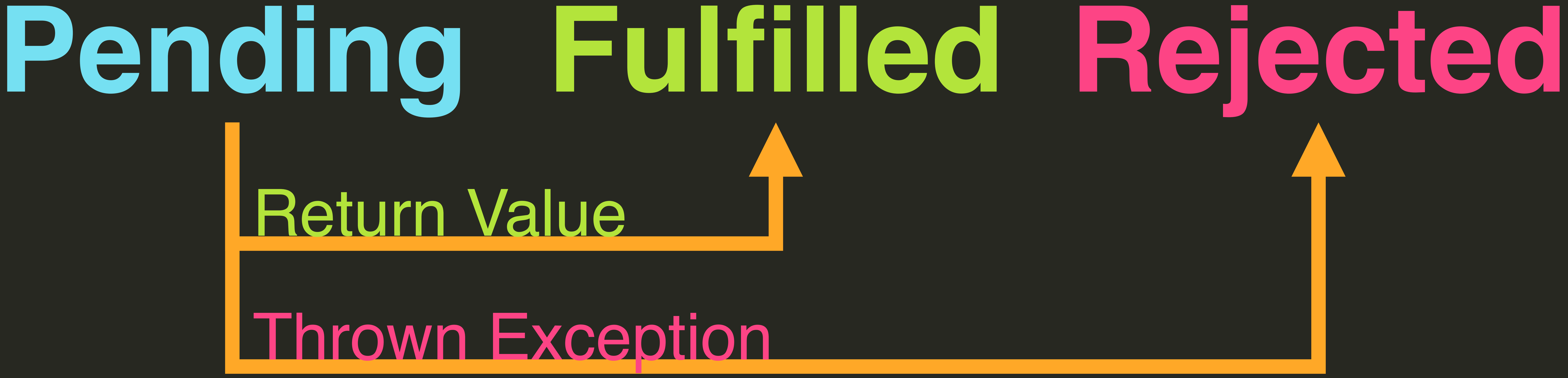
Promise States

$$\begin{tikzcd}
\text{Pending} \arrow[r, bend right] \arrow[rr, bend right=50] & \text{Fulfilled} & \text{Rejected}
\end{tikzcd}$$

Promise States



Promise States



Consuming Promises

Promise Prototype Methods

Promise.prototype.then

```
getJSON('/comments')
  .then(onFulfilled, onRejected)
```

Promise.prototype.then

```
getJSON('/comments')
  .then(function(comments) {
    if (comments) return 'Good'
    else throw new Error('Bad')
  }, function(reason) {
    // handle getJSON errors
  })
```

Promise.prototype.then

```
getJSON('/comments')
  .then(fulfillOne, rejectOne)
  .then(fulfillTwo)
  .then(null, rejectTwo)
```

Promise.prototype.then

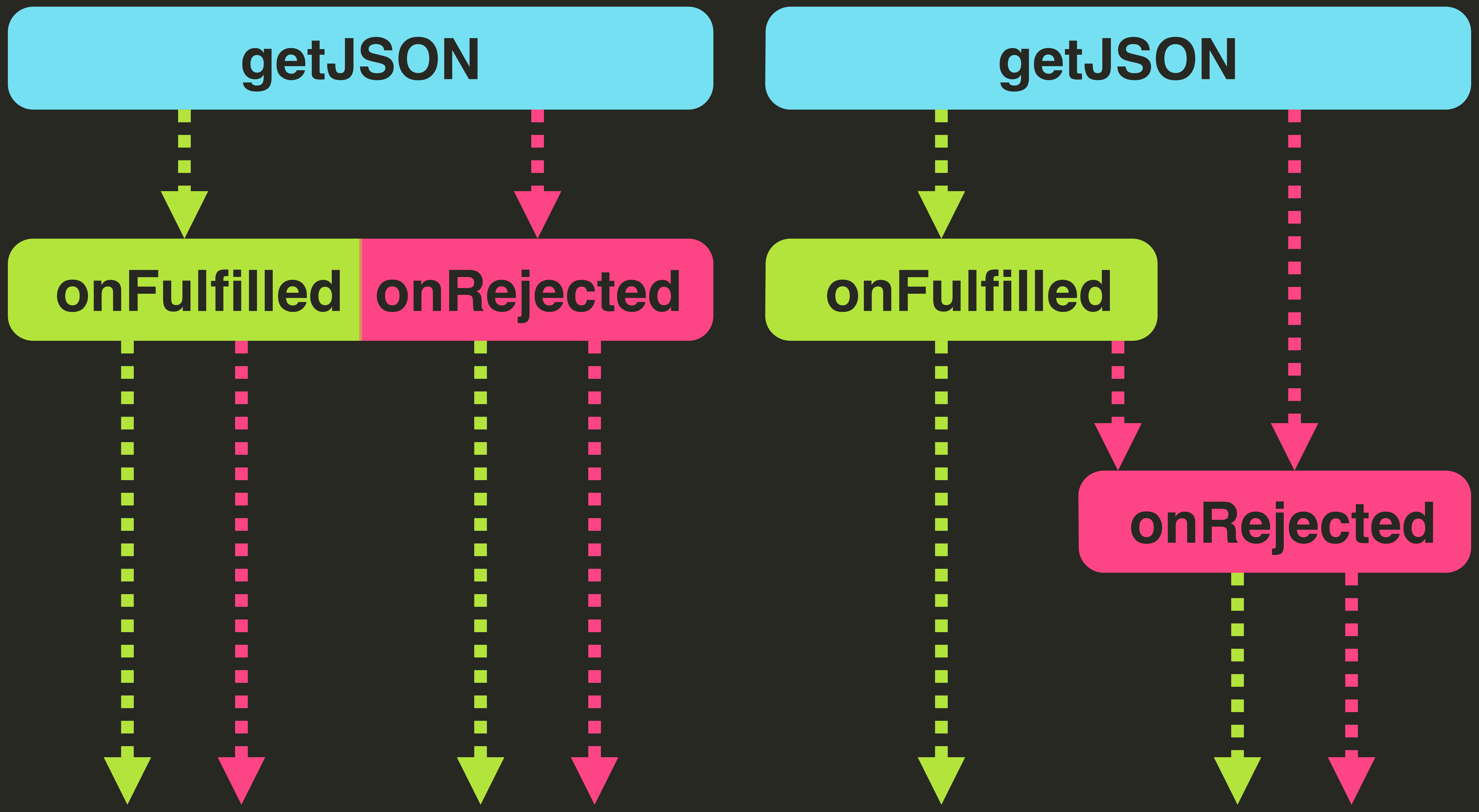
THROW

```
getJSON('/comments')
  .then(fulfillOne, rejectOne)
  .then(fulfillTwo)
  .then(null, rejectTwo)
```

Promise.prototype.then

```
getJSON('/comments')
  .then(fulfillOne, rejectOne)
  .then(fulfillTwo)
  .then(null, rejectTwo)
```

THROW


getJSON
onFulfilled
onRejected
getJSON
onFulfilled
onRejected

Promise.prototype.then

```
getJSON('/comments') // waiting...
  .then(extractFirstId)
  .then(getCommentById) // waiting...
  .then(showComment)
```

Promise.prototype.then

```
var promise = getJSON('/comments');
somethingElse();
promise.then(onFulfilled, onRejected);
```

Promise.prototype.catch

```
getJSON('/comments')
  .then(null, onRejected)
```

Promise.prototype.catch

```
getJSON('/comments')
  .catch(onRejected)
```

Promise Static Methods

Promise.cast

```
Promise.cast('Hi you!')
   .then(onFulfilled, onRejected)
```

Promise.cast

```
Promise.cast(seededRandom(42))
    .then(onFulfilled, onRejected)
```

Promise.cast

```
Promise.cast($.ajax(config))
  .then(onFulfilled, onRejected)
```

```
$.ajax(config)
  .then(onFulfilled, onRejected)
```

```
$.ajax(config)
  .then(onFulfilled, onRejected)
```

Lies, all LIES!

Promise.cast

```
Promise.cast($.ajax(config))
   .then(onFulfilled, onRejected)
```

Promise.all

```
var promises = [ getJSON('/a'),
  getJSON('/b'), getJSON('/c') ]

Promise.all(promises)
  .then(allFulfilled, firstRejected)
```

Promise.all

```
var promises = [ getJSON('/a'),
   'Hi you!', 42, getJSON('/c') ]

Promise.all(promises)
   .then(allFulfilled, firstRejected)
```

Promise.race

```
var promises = [ saveTo(server1),
  saveTo(server2), saveTo(server3) ]

Promise.race(promises)
  .then(firstFulfilled, firstRejected)
```

Creating Promises

new Promise()

new Promise()

```
function() {
  var name = prompt('Your name?')
  if (!name)
    throw new Error('Rude user!')
  else
    return name
}
```

new Promise()

```
new Promise(function(fulfill, reject) {
  var name = prompt('Your name?')
  if (!name)
    throw new Error('Rude user!')
  else
    return name
})
```

new Promise()

```
new Promise(function(fulfill, reject) {
  var name = prompt('Your name?')
  if (!name)
   reject(new Error('Rude user!'))
  else
   fulfill(name)
})
```

Shortcuts

Promise.resolve()

```
new Promise(function(fulfill, reject) {
  fulfill(something)
})
```

Promise.resolve()

```
new Promise(function(fulfill, reject) {
  fulfill(something)
})
Promise.resolve(something)
```

Promise.reject()

```
new Promise(function(fulfill, reject) {
  reject(something)
})
```

Promise.reject()

```
new Promise(function(fulfill, reject) {
  reject(something)
})
Promise.reject(something)
```

Advanced Techniques

this and bind()

```
this.get('name') // 'Kerrick'

getJSON('/comments')
  .then(function(comments) {
    this.get('name') // throws
  })
```

this and bind()

```
this.get('name') // 'Kerrick'
var self = this
getJSON('/comments')
  .then(function(comments) {
    self.get('name') // 'Kerrick'
  })
```

this and bind()

```
this.get('name') // 'Kerrick'

getJSON('/comments')
  .then(function(comments) {
    this.get('name') // 'Kerrick'
  }.bind(this))
```

Absorbed Rejections

```
getJSON('/comments')
  .then(function(comments) {
    throw new Error('Hello?')
  })
```

Absorbed Rejections

```
getJSON('/comments')
  .then(function(comments) {
    throw new Error('Hello?')
  })
  .catch(console.error.bind(console))
```

Promise-aware Functions

```
getStudent(name) // <Promise>
   .then(getLatestGrade)
   .then(log)
   .catch(logError)
   .then(logOut)
```

Promise-aware Functions

```
log(getLatestGrade(getStudent(name)))
  .catch(logError)
  .then(logOut)
```

Promise-aware Functions

```
function getLatestGrade(promise) {
  return Promise.cast(promise)
    .then(function(value) {
      // getLatestGrade Logic
    })
}
```